Tugas Kelompok

Nama Kelompok	1. Moh Naufal Faqih (10222044) 2. Firman Firdaus (10222033) 3. Ryan Azis Saputra (10222041) 4. Andika Fitra Ramadan (10222043)
Mata Kuliah	Pengolahan dan Analisis Sains Data
Dosen	Dr. Susmini Indriani Lestariningati, S.T, M.T.
Program Studi	Sistem Komputer

---

1. Deteksi Missing Value

a. Program

```python
missingValues = []

for i, row in enumerate(data):
  for j, value in enumerate(row):
    if value.strip() == '' or value.lower() == "nan" or value is None:
      missingValues.append((i + 1, header[j]))
if missingValues:
  print(f"Missing value Ditemukan pada:")
  for row, col in missingValues:
    print(f" - Baris {row}, Kolom {col}")
  print(f"Total Data yang Hilang: {len(missingValues)}")
else:
  print("Tidak Ditemukan Missing Value.")
```

b. Output

```
Missing value Ditemukan pada:
 - Baris 1, Kolom HARGA
 - Baris 1, Kolom LT
 - Baris 9, Kolom HARGA
 - Baris 19, Kolom HARGA
Total Data yang Hilang: 4
```

2. Deteksi Duplikasi Data

a. Program

```python
duplikasi_identik = {}

for i, row in enumerate(data):

    key = tuple(row)
    if key in duplikasi_identik:
        duplikasi_identik[key].append(i + 1)
    else:
        duplikasi_identik[key] = [i + 1]


duplikasi_identik = {key: value for key, value in duplikasi_identik.items() if len(value) > 1}
totalDuplikasiIdentik = sum(len(rows) for rows in duplikasi_identik.values())


if duplikasi_identik:
    print("Data identik yang terduplikasi ditemukan:")
    for key, rows in duplikasi_identik.items():
        print(f" - Data: {key} ditemukan di baris: {rows}")
    print(f"Total data identik yang terduplikasi: {totalDuplikasiIdentik}")
else:
    print("Tidak ada data identik yang terduplikasi.")
```

b. Output

```
Data identik yang terduplikasi ditemukan:
 - Data: ('7,600,000,000', '278', '350', '4', '4', 'ADA', 'JAKSEL') ditemukan di baris: [7, 109]
 - Data: ('7,000,000,000', '384', '400', '4', '4', 'ADA', 'JAKSEL') ditemukan di baris: [14, 285]
 - Data: ('5,800,000,000', '144', '285', '4', '3', 'ADA', 'JAKSEL') ditemukan di baris: [19, 318, 709]
 - Data: ('40,000,000,000', '1500', '700', '4', '4', 'ADA', 'JAKSEL') ditemukan di baris: [23, 125]
 - Data: ('5,680,000,000', '151', '245', '3', '3', 'ADA', 'JAKSEL') ditemukan di baris: [27, 544]
 - Data: ('10,500,000,000', '220', '350', '4', '3', 'ADA', 'JAKSEL') ditemukan di baris: [28, 435]
 - Data: ('7,050,000,000', '128', '315', '3', '3', 'ADA', 'JAKSEL') ditemukan di baris: [29, 823]
 - Data: ('68,000,000,000', '470', '1000', '5', '5', 'ADA', 'JAKSEL') ditemukan di baris: [32, 837]
 - Data: ('85,000,000,000', '770', '500', '4', '4', 'ADA', 'JAKSEL') ditemukan di baris: [34, 844]
 - Data: ('20,900,000,000', '754', '500', '4', '4', 'ADA', 'JAKSEL') ditemukan di baris: [36, 319]
 - Data: ('14,000,000,000', '410', '800', '4', '4', 'ADA', 'JAKSEL') ditemukan di baris: [38, 299]
 - Data: ('14,000,000,000', '460', '400', '4', '3', 'ADA', 'JAKSEL') ditemukan di baris: [71, 607]
 - Data: ('8,000,000,000', '323', '317', '5', '4', 'ADA', 'JAKSEL') ditemukan di baris: [75, 107]
 - Data: ('10,000,000,000', '395', '450', '5', '5', 'ADA', 'JAKSEL') ditemukan di baris: [79, 592]
 - Data: ('5,500,000,000', '234', '115', '2', '2', 'ADA', 'JAKSEL') ditemukan di baris: [93, 785]
 - Data: ('16,000,000,000', '407', '575', '5', '4', 'ADA', 'JAKSEL') ditemukan di baris: [122, 717]
 - Data: ('55,000,000,000', '1500', '1200', '6', '6', 'ADA', 'JAKSEL') ditemukan di baris: [128, 239]
 - Data: ('21,000,000,000', '1042', '550', '7', '6', 'ADA', 'JAKSEL') ditemukan di baris: [144, 722]
 - Data: ('22,000,000,000', '455', '570', '4', '5', 'ADA', 'JAKSEL') ditemukan di baris: [149, 228, 740]
 - Data: ('15,000,000,000', '714', '750', '5', '5', 'ADA', 'JAKSEL') ditemukan di baris: [155, 167]
 - Data: ('8,000,000,000', '380', '450', '5', '5', 'ADA', 'JAKSEL') ditemukan di baris: [161, 191]
 - Data: ('18,900,000,000', '470', '660', '4', '4', 'TIDAK ADA', 'JAKSEL') ditemukan di baris: [162, 192]
 - Data: ('21,900,000,000', '377', '260', '6', '5', 'ADA', 'JAKSEL') ditemukan di baris: [164, 193]
 - Data: ('5,600,000,000', '269', '300', '4', '4', 'ADA', 'JAKSEL') ditemukan di baris: [165, 194]
...
 - Data: ('700,000,000', '60', '52', '3', '2', 'ADA', 'JAKSEL') ditemukan di baris: [797, 814]
 - Data: ('14,900,000,000', '291', '400', '4', '4', 'TIDAK ADA', 'JAKSEL') ditemukan di baris: [802, 857]
 - Data: ('1,200,000,000', '60', '70', '3', '2', 'ADA', 'JAKSEL') ditemukan di baris: [806, 818, 890]
Total data identik yang terduplikasi: 130
```

c. Catatan Tambahan

- Deteksi pada duplikasi hanya dilakukan pada data yang benar-benar identik, dengan mempertimbangkan nilai pada kolom lainnya. Seperti luas tanah (LT), luas bangunan (LB), jumlah kamar tidur (JKT), jumlah kamar mandi (JKM), dan garasi (GRS).
- Jika deteksi duplikasi hanya dilakukan pada nilai kolom harga saja itu akan terdeteksi 735 data yang harganya mempunyai nilai yang sama.

3. Deteksi Outlier Menggunakan Metode IQR

a. Program

```python
# Deteksi Outlier
def hitungIQR(data):
    data.sort()
    n = len(data)
    q1 = data[n // 4] if n % 4 == 0 else (data[n // 4] + data[n // 4 - 1]) / 2
    q3 = data[3 * n // 4] if (3*n) % 4 == 0 else (data[3 * n // 4] + data[3*n // 4 - 1]) / 2
    iqr = q3 - q1
    return q1, q3, iqr

dataHarga = [float(row[0].replace(',', '')) for row in data if row[0].strip() != '' and row[0].lower() != 'nan']
q1, q3, iqr = hitungIQR(dataHarga)

lowerBound = q1 - 1.5 * iqr
upperBound = q3 + 1.5 * iqr

outliers = [value for value in dataHarga if value < lowerBound or value > upperBound]
totalOutliers = len(outliers)
print(f"Total Outlier: {totalOutliers}")

print(f"Q1: {q1}, Q3: {q3}, IQR: {iqr}")
print(f"batas bawah: {lowerBound}, batas atas: {upperBound}")
if outliers:
    print(f"Outlier ditemukan: {outliers}")
else:
    print("Tidak ada outlier ditemukan")
```

b. Output

Total Outlier: 65
Q1: 6750000000.0, Q3: 20000000000.0, IQR: 13250000000.0
batas bawah: -13125000000.0, batas atas: 39875000000.0
Outlier ditemukan: [40000000000.0, 40000000000.0, 40000000000.0, 40000000000.0, 40000000000.0, 40000000000.0, 40000000000.0, 40000000000.0, 42000000000.0, 42000000000.0, 42000000000.0, 43000000000.0, 43000000000.0, 45000000000.0, 45000000000.0, 45000000000.0, 47000000000.0, 49000000000.0, 50000000000.0, 50000000000.0, 55000000000.0, 55000000000.0, 55000000000.0, 55000000000.0, 55000000000.0, 55000000000.0, 55000000000.0, 55000000000.0, 55000000000.0, 56000000000.0, 57000000000.0, 60000000000.0, 60000000000.0, 60000000000.0, 60000000000.0, 65000000000.0, 65000000000.0, 68000000000.0, 68000000000.0, 68000000000.0, 68000000000.0, 72000000000.0, 75000000000.0,

75000000000.0, 75000000000.0, 77500000000.0, 80000000000.0,
85000000000.0, 85000000000.0, 85000000000.0, 85000000000.0,
85500000000.0, 90000000000.0, 95000000000.0, 100000000000.0,
110000000000.0, 120000000000.0, 135000000000.0, 165000000000.0,
168000000000.0, 169000000000.0, 175230000000.0, 180000000000.0,
185000000000.0, 250000000000.0]

c. Catatan Tambahan:

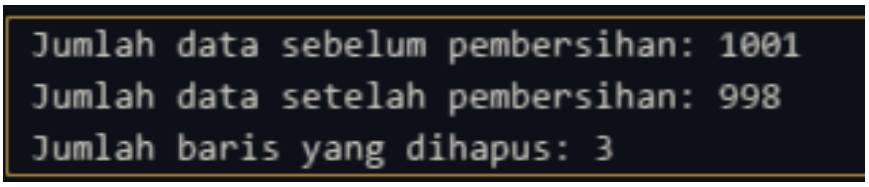
- Total outlier sebanyak 65 data terdeteksi sebagai outlier.
- Ini menunjukkan bahwa ada banyak data yang berada di luar batas distribusi normal harga rumah.
- Q1 (Kuartil Pertama) 6.75 miliar (6.750.000.000.0) menunjukkan 25% dari data memiliki harga rumah di bawah nilai ini.
- Q3 (Kuartil Ketiga) 20 miliar (20.000.000.000.0) menunjukkan 75% dari data memiliki harga rumah di bawah nilai ini.
- IQR (Interquartile Range) 13.25 miliar (13.250.000.000.0) menunjukkan rentang variasi harga rumah antara Q1 dan Q3 di tengah distribusi.
- Harga rumah yang sangat tinggi (di atas 39.875 miliar) kemungkinan besar mencerminkan properti mewah atau lokasi premium dengan mempertimbangkan faktor faktor lainnya, seperti luas tanah, luas bangunan, jumlah kamar tidur, jumlah kamar mandi, dan garasi.

4. Proses Menangani Missing Value

a. Program

```
cleaned_data = [row for row in data if all(value.strip() != "" and value.lower() != "nan" and value is not None for value in row)]

# Output hasil
print(f"Jumlah data sebelum pembersihan: {len(data)}")
print(f"Jumlah data setelah pembersihan: {len(cleaned_data)}")
print(f"Jumlah baris yang dihapus: {len(data) - len(cleaned_data)}")
```

b. Output



```
Jumlah data sebelum pembersihan: 1001
Jumlah data setelah pembersihan: 998
Jumlah baris yang dihapus: 3
```

c. Catatan Tambahan

- Karena data yang missing/hilang hanya sedikit yaitu 3, melakukan penghapusan pada data adalah keputusan yang tepat dibanding melakukan mengisi (imputasi) yang bisa menambah bias atau error.

- Karena data hanya 0.3% yang dihapus dari seluruh data maka tidak akan memengaruhi rata-rata, median maupun sebaran data.

5. Proses Menangani Duplikasi Data

a. Program

```
dataUnik = []
seen = set()

for row in cleaned_data:
    rowTuple = tuple(row)
    if rowTuple not in seen:
        seen.add(rowTuple)
        dataUnik.append(row)
print(f"Jumlah data sebelum penghapusan duplikasi: {len(cleaned_data)}")
print(f"Jumlah data setelah penghapusan duplikasi: {len(dataUnik)}")
print(f"Jumlah baris yang dihapus: {len(cleaned_data) - len(dataUnik)}")

```

b. Output

```
Jumlah data sebelum penghapusan duplikasi: 998
Jumlah data setelah penghapusan duplikasi: 929
Jumlah baris yang dihapus: 69
```

6. Proses Menangani Outlier

a. Program

```
dataHarga = [float(row[0].replace(',', '')) for row in dataUnik if row[0].strip() != '' and row[0].lower() != 'nan']

# Hitung IQR
q1, q3, iqr = hitungIQR(dataHarga)
lowerBound = q1 - 1.5 * iqr
upperBound = q3 + 1.5 * iqr

segmentasiData = []
for row in dataUnik:
    harga = float(row[0].replace(',', ''))
    if harga > lowerBound and harga < iqr:
        row.append("Harga Rumah di Bawah Harga Reguler")
    elif harga > iqr and harga < upperBound:
        row.append("Rumah Reguler")
    elif harga > upperBound:
        row.append("Rumah Mewah")
    segmentasiData.append(row)
print(f"Jumlah rumah di bawah Harga Reguler: {sum(1 for row in segmentasiData if row[-1] == 'Harga Rumah di Bawah Harga Reguler')}")
print(f"Jumlah rumah reguler: {sum(1 for row in segmentasiData if row[-1] == 'Rumah Reguler')}")
print(f"Jumlah rumah mewah: {sum(1 for row in segmentasiData if row[-1] == 'Rumah Mewah')}")

```

b. Output

```
Jumlah rumah di bawah Harga Reguler: 467
Jumlah rumah reguler: 402
Jumlah rumah mewah: 60
```

c. Catatan Tambahan

- Outlier pada data ditandai dengan membaginya ke beberapa segmen berdasarkan harganya (misalnya: di bawah harga reguler dan mewah).
- Ada beberapa rumah dengan harga sangat tinggi yang kemungkinan memiliki fasilitas eksklusif dan lengkap.
- Outlier tidak dianggap sebagai "kesalahan" atau "anomali", tapi dijadikan kategori khusus yang memberi makna baru dalam analisis.
- Dengan mengelompokkan harga ekstrem ke segmen tersendiri, dapat menghindari distorsi pada analisis rata-rata, atau median, karena outlier tak lagi mempengaruhi data utama

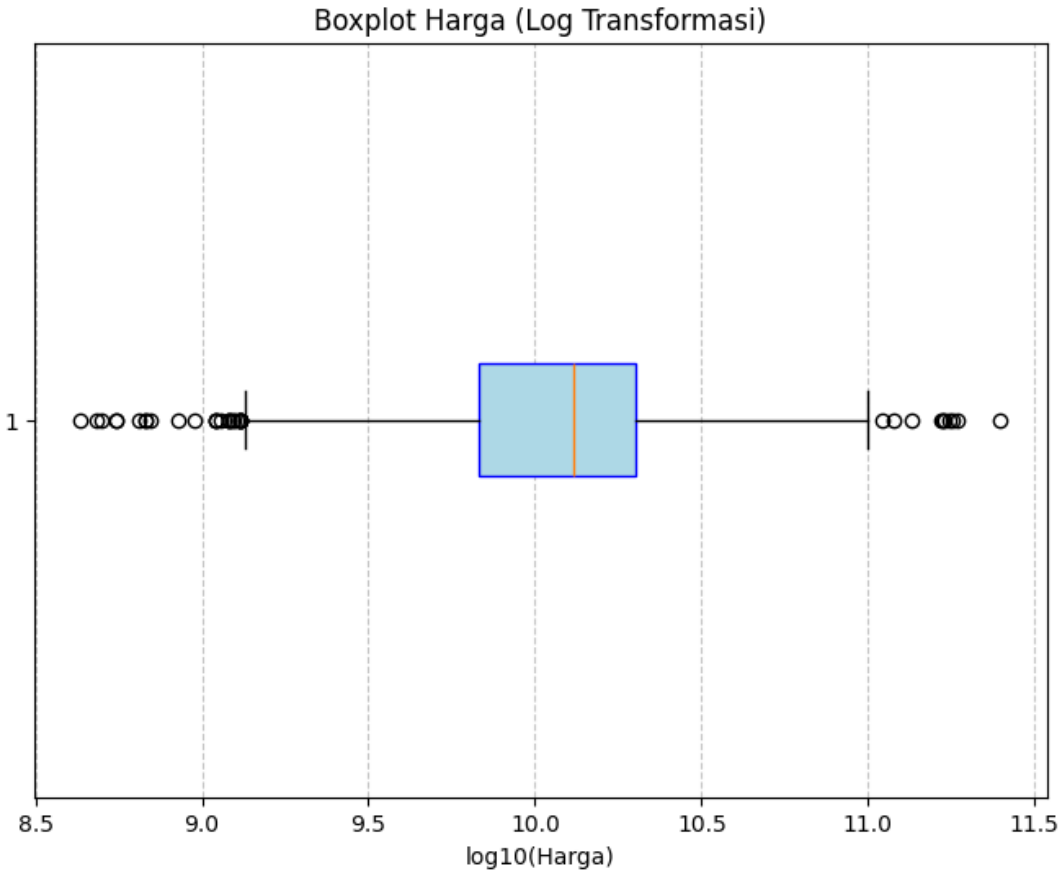
7. Visualisasi Data (Boxplot dan Histogram)

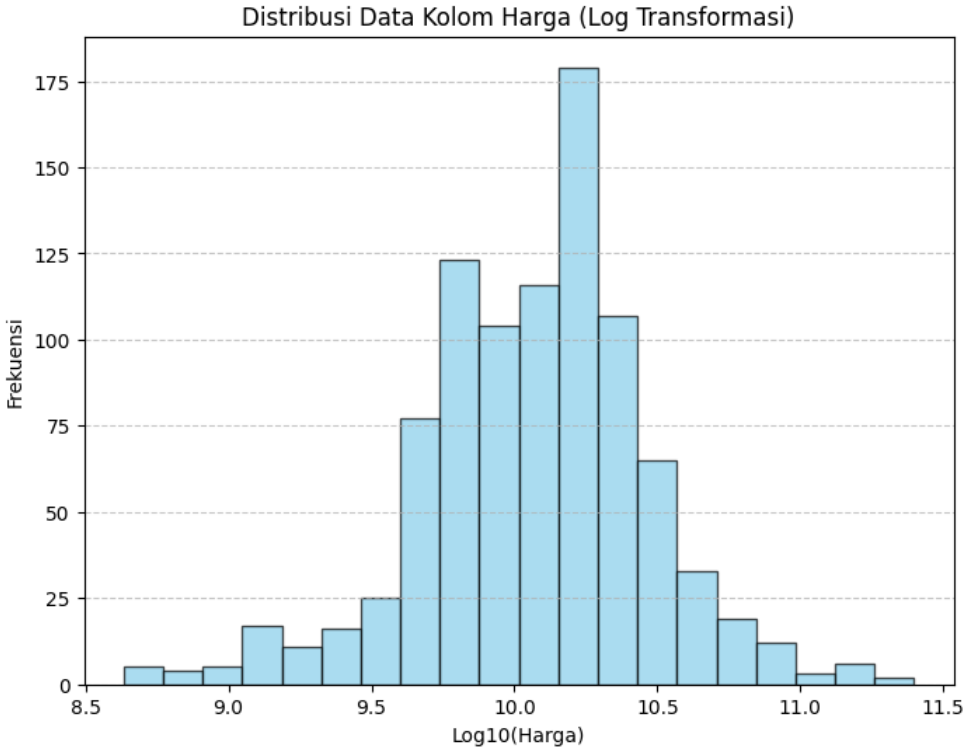
a. Program

```python
# visualisasi data( boxplot)
log_data_harga = np.log10(dataHarga)
plt.figure(figsize=(8, 6))
plt.boxplot(log_data_harga, vert=False, patch_artist=True, boxprops=dict(facecolor='lightblue', color='blue'))
plt.title("Boxplot Harga (Log Transformasi)")
plt.xlabel("log10(Harga)")
plt.grid(axis='x', linestyle='--', alpha=0.7)

# Tampilkan plot
plt.show()
```

```python
log_data_harga = np.log10(dataHarga)

# Membuat histogram
plt.figure(figsize=(8, 6))
plt.hist(log_data_harga, bins=20, color='skyblue', edgecolor='black', alpha=0.7)
plt.title("Distribusi Data Kolom Harga (Log Transformasi)")
plt.xlabel("Log10(Harga)")
plt.ylabel("Frekuensi")
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Tampilkan plot
plt.show()
```

b. Output


Boxplot Harga (Log Transformasi)
1
8.5
9.0
9.5
10.0
10.5
11.0
11.5
log10(Harga)


Distribusi Data Kolom Harga (Log Transformasi)
Frekuensi
0
25
50
75
100
125
150
175
8.5
9.0
9.5
10.0
10.5
11.0
11.5
Log10(Harga)

c. Catatan Tambahan

- Nilai pada harga rumah awalnya condong ke kiri, artinya Sebagian besar harga ada di level rendah-menengah tapi ada segelintir yang sangat mahal.
- Maka dari itu dilakukanlah transformasi logaritmik (misal: log10(harga)) menormalkan distribusi, membuatnya lebih mendekati distribusi normal sehingga lebih mudah dianalisis secara statistik.
- Tanpa log transform, data mahal bisa menutupi variasi data di harga rendah.
- Dengan log, semua nilai bisa terlihat lebih proporsional, sehingga struktur sebaran data menjadi lebih jelas (seperti kelompok harga tertentu atau outlier)
- Log transformasi menekan nilai-nilai ekstrem (contohnya rumah 200 miliar) sehingga outlier tidak terlalu mendistorsi skala grafik.